Question 1

Design behaviour trees for the following NPCs.

a) Two guards patrol a number of rooms together, sometimes waiting in a room before moving on to the next. If they spot the player then they both attack, although guard 2 first locates and sets off the wall-mounted security alarm in that room.

b) The cleaner robot from the lecture, as specified by the hierarchical state machine. Give two sub-trees for the recharge and cleanup behaviours, co-ordinated by a third main tree.
Solution 1b
Main Tree

Repeat this tree:

- **Low energy?**
  - **Until success**
    - **Interrupt**
    - **Recharge**
  - **Interrupter**
  - **Cleanup**

**Solution 1b**

**Recharge**

- **Move to station**
  - **Until success**
  - **Charging**
  - **Full power?**

**Cleanup**

- **Repeat**
  - **?**
    - **Trash visible?**
      - **Trash**
      - **Collect trash**
    - **Have trash?**
      - **Move to bin**
      - **Dispose trash**
    - **Near bin?**
      - **Wait**
Question 2

a) A player can move L(eft) and R(ight) in game of Space Invaders. Construct a hierarchical 4-gram table based on the input LLLRL RLRLL LLRLR LRRRL LLRRR LRLRR.

b) Predict the next action for all possible inputs of length 3, using a sample threshold of 5. Give the n-value used and probability for each prediction.

Solution 2

<table>
<thead>
<tr>
<th>2 x 1-Grams Sample 30</th>
<th>8 x 3-Grams Sample 28</th>
<th>16 x 4-Grams Sample 27</th>
<th>Predictions Threshold 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>L 16</td>
<td>LLL 4</td>
<td>LLLL 1</td>
<td>LLL L 3</td>
</tr>
<tr>
<td>R 14</td>
<td>LLR 3</td>
<td>LLLR 3</td>
<td>LLR L 3</td>
</tr>
<tr>
<td></td>
<td>LRL 6</td>
<td>LLRL 2</td>
<td>LRL L 3</td>
</tr>
<tr>
<td></td>
<td>LRR 3</td>
<td>LLRR 1</td>
<td>LRR L 2</td>
</tr>
<tr>
<td></td>
<td>RLL 2</td>
<td>LRLL 1</td>
<td>RLL L 1</td>
</tr>
<tr>
<td></td>
<td>RLR 6</td>
<td>LRRR 1</td>
<td>RRR L 2</td>
</tr>
<tr>
<td></td>
<td>RRL 2</td>
<td>LRRL 5</td>
<td>LRRL L 1</td>
</tr>
<tr>
<td></td>
<td>RRR 2</td>
<td>LRLL 0</td>
<td>LRLL R 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LRRR 2</td>
<td>LRRR R 2</td>
</tr>
</tbody>
</table>

Predictions:

<table>
<thead>
<tr>
<th>Next</th>
<th>n</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LLL</td>
<td>3</td>
<td>0.57</td>
</tr>
<tr>
<td>LLR</td>
<td>3</td>
<td>0.67</td>
</tr>
<tr>
<td>LRL</td>
<td>4</td>
<td>0.83</td>
</tr>
<tr>
<td>LRR</td>
<td>2</td>
<td>0.61</td>
</tr>
<tr>
<td>RLL</td>
<td>3</td>
<td>0.57</td>
</tr>
<tr>
<td>RLR</td>
<td>4</td>
<td>0.67</td>
</tr>
<tr>
<td>RRR</td>
<td>2</td>
<td>0.75</td>
</tr>
<tr>
<td>RRR</td>
<td>2</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Question 3

An RTS game agent bases its strategy its strength and territory relative to the player, using the following model of the game

States: SL (strong-large), SS (strong-small), WL (weak-large), WS (weak-small)

Transitions: A(attack), E xplore), R(etreat), B(uild) (The state graph is complete.)

The agent uses Q-learning to adapt to a new player. All Q-values start at 1, $\alpha = \gamma = 0.75$.

a) Describe the Q function after the experiences <WS, B, 0.5, SS>, <SS, E, 0.6, SL>, <SL, A, 1.0, SL>, <SL, R, -1.0, SS>

b) Why are these initial Q-values unsatisfactory?

c) Why might you want to change $\alpha$ as training progresses

Solution 3a

If $<s, a, r, s'>$ then $Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max(Q(s', a')))$

$<\text{WS, B, 0.5, SS}>$ $Q(\text{WS, B}) = 0.25 \times 1 + 0.75(0.5 + 0.75 \times 1) = 1.19$

$<\text{SS, E, 0.6, SL}>$ $Q(\text{SS, E}) = 0.25 \times 1 + 0.75(0.6 + 0.75 \times 1) = 1.26$

$<\text{SL, A, 1.0, SL}>$ $Q(\text{SL, A}) = 0.25 \times 1 + 0.75(1 + 0.75 \times 1) = 1.56$

$<\text{SL, R, -1.0, SS}>$ Note that $\max(Q(\text{SS, } *)) = Q(\text{SS, E}) = 1.26$ so we get $Q(\text{SL, R}) = 0.25 \times 1 + 0.75(-1 + 0.75 \times 1.26) = 0.21$

So after the first four experience tuples all Q-values are still 1, except for the four values given above.
b) Initially all actions are valued equally in all states, which will make the agent behave randomly until it adapts. It might be better to start with Q-values that reflect a generic strategy before adapting it to a new player.

c) The learning rate $\alpha$ decides how much new experiences are favoured over old. A high $\alpha$ is useful initially to kick start learning. But if our player is unlikely to change then we might want to lower $\alpha$ as learning progresses. This will protect our accumulated experience, i.e. we will require a greater amount of evidence to change our strategy.